Numerical Algorithms (2021) 87:1061–1074
https://doi.org/10.1007/s11075-020-00998-3

ORIGINAL PAPER

The complex step approximation to the higher order Fréchet derivatives of a matrix function

Awad H. Al-Mohy[1] · **Bahar Arslan**[2]

Received: 29 December 2019 / Accepted: 4 August 2020 / Published online: 31 August 2020

Abstract

The kth Fréchet derivative of a matrix function f is a multilinear operator from a cartesian product of k subsets of the space $\mathbb{C}^{n\times n}$ into itself. We show that the kth Fréchet derivative of a real-valued matrix function f at a real matrix A in real direction matrices $E_1, E_2, \ldots, E_k$ can be computed using the complex step approximation. We exploit the algorithm of Higham and Relton (*SIAM J. Matrix Anal. Appl.* 35(3):1019–1037, 2014) with the complex step approximation and mixed derivative of complex step and central finite difference scheme. Comparing with their approach, our cost analysis and numerical experiment reveal that *half* and *seven-eighths* of the computational cost can be saved for the complex step and mixed derivative, respectively. When f has an algorithm that computes its action on a vector, the computational cost drops down significantly as the dimension of the problem and k increase.

Keywords Matrix function · Fréchet derivative · Higher order Fréchet derivative · Complex step approximation · Action of matrix functions

1 Introduction

Matrix functions play important roles in a variety of applications such as quantum graphs [10], network analysis [11], computer animation [29], and solutions of systems of differential equations [3, 7]. In the computation of matrix functions, it is

✉ Awad H. Al-Mohy
ahalmohy@kku.edu.sa

Bahar Arslan
bahar.arslan@btu.edu.tr

[1] Department of Mathematics, King Khalid University, Abha, Saudi Arabia

[2] Mathematics Department, Faculty of Engineering and Natural Sciences, Bursa Technical University, Yıldırım/Bursa, Turkey

important to understand how perturbations in input data effect the results. The condition number plays an essential role in measuring the sensitivity of the data to perturbations and the norm of the Fréchet derivative is the main component of the condition number. Apart from the sensitivity analysis, the Fréchet derivative is also used in image reconstruction in tomography [28], the computation of choice probabilities [2], the analysis of carcinoma treatment [14], and computing the matrix geometric mean and Karcher mean [19]. In the literature, there are some numerical algorithms for the Fréchet derivative for the matrix exponential, square root, logarithm, and fractional power; see [4, 8, 12, 15, 17, 21]. By using Daleckiǐ-Kreǐn formula, Noferini gives an explicit expression for the Fréchet derivative of generalized matrix functions [27].

The second-order Fréchet derivative has an application in the extension of iterative methods to solve a nonlinear scalar equation to Banach spaces [9]. The computation of the higher order Fréchet derivative s of matrix functions was first proposed by Higham and Relton [18]. The authors develop algorithms for computing the kth derivative and its Kronecker form. They analyze the level-2 absolute condition number of a matrix function, which is the condition number of the condition number, and bound it in terms of the norm of the second Fréchet derivative.

The use of complex arithmetic to approximating the derivative of analytic functions was introduced by Lyness [26] and Lyness and Moler [25]. The work of Squire and Trapp [30] appears the earliest manipulation of the complex step approximation for the derivative of a real function at a real number. Later, many authors have extended the complex step approach to produce approximations of higher rate of convergence (see, for instance, [1, 22–24]). The extension of the complex step approximation to the first-order Fréchet derivative of real matrix functions was first introduced by Al-Mohy and Higham [6].

The aim of this work is to make the computation of the higher order Fréchet derivative of a matrix function as efficient as possible via the use of derivative techniques: complex step and finite difference, and the implementation of the action of the matrix function on a thin tall matrix whenever available. The paper is organized as follows. In Section 2 we review the definition and the computation aspects of the kth Fréchet derivative given by Higham and Relton [18]. Based on the definition, we derive a recurrence relation for the kth Fréchet derivative of a monomial matrix function. Section 3 represents the main body of the paper. First, using the definition of the kth Fréchet derivative, we derive the central finite difference scheme and the complex step approximation showing the order of convergence rate. Second, we use the block matrix X_{k-1} (half the size of X_k) alongside the complex step approximation to compute the kth Fréchet derivative, yielding about 50% saving of the computational cost. Third, we derive the mixed derivative scheme of the central finite difference and the complex step approximation. This allowed us to use the block matrix X_{k-2} (one-fourth the size of X_k) with certain inputs to compute the kth Fréchet derivative. The computational saving is about 87%. Fourth, since the kth Fréchet derivative is extracted from $f(X_k)$ by reading off the top right $n \times n$ block, it is attractive to use the action of $f(X_k)$ on a certain thin and tall matrix to extract that block. We explain how to obtain the kth Fréchet derivative as a whole matrix and how to obtain its action on a vector. This approach yields a significant reduction of computational cost and CPU time. Finally, we give our numerical experiment in Section 4 and draw our concluding remarks in the last section.

2 Higher order Fréchet derivative

The kth-order Fréchet derivative of $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at $A \in \mathbb{C}^{n\times n}$ can be defined recursively as the unique multilinear operator $L_f^{(k)}(A)$ of the direction matrices $E_i \in \mathbb{C}^{n\times n}$, $i = 1: k$, that satisfies

$$\|L_f^{(k-1)}(A+E_k, E_1, \cdots, E_{k-1}) - L_f^{(k-1)}(A, E_1, \cdots, E_{k-1}) - L_f^{(k)}(A, E_1, \cdots, E_k)\| = o(\|E_k\|), \quad (2.1)$$

where $L_f^{(0)}(A) = f(A)$ and $L_f^{(1)}(A, E_1)$ is the first-order Fréchet derivative. To shorten our expressions, we denote the k-tuple $(E_1, E_2, \cdots, E_k)$ by $\mathcal{E}_k$ regardless of the order of E_k since the multilinear operator $L_f^{(k)}(A)$ is symmetric.

If $E = E_j$, $j = 1: k$, we denote the kth Fréchet derivative of f at E by $L_f^{(k)}(A, E)$; that is, $L_f^{(k)}(A, E) = L_f^{(k)}(A, E, E, \cdots, E)$.

For the monomial X^r, where r is any nonnegative integer, we obtain the following recurrence for $L_{x^r}^{(k)}(A, \mathcal{E}_k)$, which can be proven by induction on k and using the product rule of the Fréchet derivative.

Lemma 2.1 *The kth Fréchet derivative of X^r is given by*

$$L_{x^r}^{(k)}(A, E_1, E_2, \cdots, E_k) = AL_{x^{r-1}}^{(k)}(A, E_1, E_2, \cdots, E_k) + \sum_{j=1}^{k} E_j L_{x^{r-1}}^{(k-1)}(A, E_1, \cdots, E_{j-1}, E_{j+1}, \cdots, E_k) \quad (2.2)$$

with $L_{x^r}^{(k)}(A, E_1, E_2, \cdots, E_k) = 0$ if $k > r$.

In particular if $E = E_j$, $j = 1: k$, the recurrence relation (2.2) boils down to the recurrence relation [6, Eq. (3.1)].

We recall next an important result by Higham and Relton [18] that allows the computation of the kth Fréchet derivative as a block of $f(X_k)$, where

$$X_k = I_2 \otimes X_{k-1} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes I_{2^{k-1}} \otimes E_k, \quad X_0 = A. \quad (2.3)$$

The symbol $\otimes$ denotes the Kronecker product [15, Chap. 12] and I_m denotes the $m \times m$ identity matrix.

We will state the following theorem for the existence of the kth Fréchet derivative.

Theorem 2.1 ([18, Theorem. 3.5]) *Let $A \in \mathbb{C}^{n\times n}$ whose largest Jordan block is of size p and whose spectrum lies in an open subset $\mathcal{D} \subset \mathbb{C}$. Let $f : \mathcal{D} \to \mathbb{C}$ be $2^k p - 1$*

times continuously differentiable on $\mathcal{D}$. Then the kth Fréchet derivative $L_f^{(k)}(A)$ exists and $L_f^{(k)}(A, \mathcal{E}_k)$ is continuous in A and $E_1, E_2, \ldots, E_k \in \mathbb{C}^{n\times n}$. Moreover

$$L_f^{(k)}(A, \mathcal{E}_k) = [f(X_k)]_{1n},$$

the upper right $n \times n$ block of $f(X_k)$.

3 Complex step approximation

3.1 Scalar case

If $f(x)$ is a real function with real variables and is analytic then it can be expanded in a Taylor series

$$f(x + \mathrm{i}h) = f(x) + \mathrm{i}hf'(x) - h^2\frac{f''(x)}{2!} - \mathrm{i}h^3\frac{f^{(3)}(x)}{3!} + \cdots . \tag{3.1}$$

Thus, the imaginary part $\mathrm{Im}(f(x + \mathrm{i}h))/h$ and the real part $\mathrm{Re}(f(x + \mathrm{i}h))$ give order $O(h^2)$ approximations to $f'(x)$ and $f(x)$, respectively. Approximating $f'(x)$ by the imaginary part of the function avoids the subtractive cancellation occurring in finite difference scheme. In addition, complex step approximation allows the use of an arbitrary small h without sacrificing the accuracy. Numerical results obtained by numerical algorithm design in meteorology proved the accuracy obtained even with $h = 10^{-100}$ [13].

Next we investigate how to implement complex step techniques to compute higher order Fréchet derivative s of matrix functions.

3.2 Matrix case

Assume that A and E_i, $i = 1{:}\ k$, are real matrices and f is a real-valued function at real arguments obeying the assumption of Theorem 2.1. Replacing the matrix E_k in the definition of the kth Fréchet derivative (2.1) by hE_k, where h is a positive real number, and exploiting the linearity of the operator $L_f^{(k)}(A)$, we have

$$L_f^{(k-1)}(A + hE_k, \mathcal{E}_{k-1}) - L_f^{(k-1)}(A, \mathcal{E}_{k-1}) - hL_f^{(k)}(A, \mathcal{E}_k) = o(h).$$

Thus

$$\lim_{h\to 0} \frac{L_f^{(k-1)}(A + hE_k, \mathcal{E}_{k-1}) - L_f^{(k-1)}(A, \mathcal{E}_{k-1})}{h} = L_f^{(k)}(A, \mathcal{E}_k), \tag{3.2}$$

which is an $O(h)$ finite difference approximation to $L_f^{(k)}(A, \mathcal{E}_k)$. The central finite difference approximation can be derived by replacing h in (3.2) by $-h$ and adding the obtained limit to (3.2) to cancel out the term $L_f^{(k-1)}(A, \mathcal{E}_{k-1})$. That is,

$$L_f^{(k)}(A, \mathcal{E}_k) = \lim_{h\to 0} \frac{L_f^{(k-1)}(A + hE_k, \mathcal{E}_{k-1}) - L_f^{(k-1)}(A - hE_k, \mathcal{E}_{k-1})}{2h}, \tag{3.3}$$

which yields $O(h^2)$ approximation to $L_f^{(k)}(A, \mathcal{E}_k)$ as shown below.

Now replacing the matrix E_k in the definition of the kth Fréchet derivative (2.1) by $\mathrm{i}hE_k$ yields

$$L_f^{(k-1)}(A + \mathrm{i}hE_k, \mathcal{E}_{k-1}) - L_f^{(k-1)}(A, \mathcal{E}_{k-1}) - \mathrm{i}hL_f^{(k)}(A, \mathcal{E}_k) = o(h).$$

Since $L_f^{(k)}(A, \mathcal{E}_k)$ is real, we obtain

$$\lim_{h\to 0} \frac{\mathrm{Im}\big(L_f^{(k-1)}(A + \mathrm{i}hE_k, \mathcal{E}_{k-1})\big)}{h} = L_f^{(k)}(A, \mathcal{E}_k). \tag{3.4}$$

This yields the complex step approximation of $L_f^{(k)}(A)$ via $L_f^{(k-1)}(A)$ as

$$L_f^{(k)}(A, \mathcal{E}_k) \approx \frac{\mathrm{Im}\big(L_f^{(k-1)}(A + \mathrm{i}hE_k, \mathcal{E}_{k-1})\big)}{h}, \tag{3.5}$$

for a sufficiently small scalar h. Clearly,

$$L_f^{(k-1)}(A, \mathcal{E}_{k-1}) \approx \mathrm{Re}\big(L_f^{(k-1)}(A + \mathrm{i}hE_k, \mathcal{E}_{k-1})\big).$$

However, this derivation of complex step approximation does not reveal the rate of convergence of the approximation as h goes to zero. To determine the rate of convergence, we need stronger assumptions on f.

Theorem 3.1 *Let $A, E_i \in \mathbb{R}^{n\times n}$, $i = 1\colon k$, and $f : \mathcal{D} \subset \mathbb{C} \to \mathbb{C}$ be an analytic function in an open subset $\mathcal{D}$ containing the spectrum of A. Assume further that f is real-valued at real arguments. Let h be a sufficiently small real number such that the spectrum of $A + \mathrm{i}hE_k$ lies in $\mathcal{D}$. Then we have*

$$L_f^{(k)}(A, \mathcal{E}_k) = \frac{\mathrm{Im}\big(L_f^{(k-1)}(A + \mathrm{i}hE_k, \mathcal{E}_{k-1})\big)}{h} + O(h^2) \tag{3.6}$$

$$L_f^{(k-1)}(A, \mathcal{E}_{k-1}) = \mathrm{Re}\big(L_f^{(k-1)}(A + \mathrm{i}hE_k, \mathcal{E}_{k-1})\big) + O(h^2) \tag{3.7}$$

where $\mathcal{E}_k = (E_1, E_2, \cdots, E_k)$.

Proof The analyticity of f on $\mathcal{D}$ implies that f has a power series expansion there. Recall [6, Theorem. 3.1] and denote by $E_k^{(j)}$ the j-tuple $(E_k, E_k, \cdots, E_k)$, so we have

$$\begin{aligned} f(A + \mathrm{i}hE_k) &= \sum_{j=0}^{\infty} \frac{(\mathrm{i}h)^j}{j!} L_f^{(j)}(A, E_k) \\ &= f(A) + \mathrm{i}hL_f(A, E_k) - \frac{h^2}{2} L_f^{(2)}(A, E_k^{(2)}) + O(h^3). \end{aligned} \tag{3.8}$$

Since the power series converges uniformly on $\mathcal{D}$, we can repeatedly Fréchet differentiate the series (3.8) term by term in the directions $E_1, E_2, \ldots, E_{k-1}$ and obtain

$$\begin{aligned} L_f^{(k-1)}(A + \mathrm{i}hE_k, \mathcal{E}_{k-1}) &= L_f^{(k-1)}(A, \mathcal{E}_{k-1}) + \sum_{j=1}^{\infty} \frac{(\mathrm{i}h)^j}{j!} L_f^{(j+k-1)}(A, E_k^{(j)}, \mathcal{E}_{k-1}) \\ &= L_f^{(k-1)}(A, \mathcal{E}_{k-1}) + \mathrm{i}hL_f^{(k)}(A, \mathcal{E}_k) \\ &- \frac{h^2}{2} L_f^{(k+1)}(A, E_{k-1}^{(2)}, \mathcal{E}_{k-1}) + O(h^3). \end{aligned}$$

Equations (3.6) and (3.7) follow immediately by equaling the imaginary and real parts of the series, respectively. □

The rate of convergence of the central finite difference approximation (3.3) can now be shown from the expansion of $L_f^{(k-1)}(A + \mathrm{i}hE_k, \mathcal{E}_{k-1})$ above by replacing the scalar h by $\mathrm{i}h$ and then by $-\mathrm{i}h$. Subtracting the expansion of $L_f^{(k-1)}(A - hE_k, \mathcal{E}_{k-1})$ from the expansion of $L_f^{(k-1)}(A + hE_k, \mathcal{E}_{k-1})$ and dividing through by $2h$ yield the approximation in (3.3) and reveal its order, $O(h^2)$. The benefit of this analysis is that when an algorithm is available to compute $L_f^{(k-1)}(A, \mathcal{E}_{k-1})$, it can be used to compute $L_f^{(k)}(A, \mathcal{E}_k)$. Thus, we can use the complex step approximation to compute $L_f^{(k)}(A, \mathcal{E}_k)$ via $\mathrm{Im} f(X_{k-1})/h$ with $X_0 = A + \mathrm{i}hE_k$ in (2.3). This leads to an important result analogous to Theorem 2.1.

Theorem 3.2 *Let $A, E_i \in \mathbb{R}^{n\times n}$, $i = 1\colon k$, and $\mathcal{D} \subset \mathbb{C}$ be an open subset containing the spectrum of A. Let h be a sufficiently small real number such that the spectrum of $A + \mathrm{i}hE_k$ lies in $\mathcal{D}$ and p_k be the size of the largest Jordan block of $A + \mathrm{i}hE_k$. Let $f : \mathcal{D} \to \mathbb{C}$ be $2^{k-1}p_k - 1$ times continuously differentiable on $\mathcal{D}$. Assume further that f is real-valued at real arguments. Then the kth Fréchet derivative exists and*

$$L_f^{(k)}(A, \mathcal{E}_k) = \lim_{h\to 0} \frac{\mathrm{Im}[f(X_{k-1})]_{1n}}{h}, \tag{3.9}$$

where $X_0 = A + \mathrm{i}hE_k$ given in (2.3).

Proof In view of Theorem 2.1, The upper right $n \times n$ block of $f(X_{k-1})$, where $X_0 = A + \mathrm{i}hE_k$, is $L_f^{(k-1)}(A + \mathrm{i}hE_k, \mathcal{E}_{k-1})$. By (3.6), $L_f^{(k)}(A, \mathcal{E}_k)$ is the upper right $n \times n$ block of $\lim_{h\to 0} \mathrm{Im} f(X_{k-1})/h$. □

The advantage of this approach is that the size of the matrix X_{k-1} is half the size of X_k, which could lead to a faster computation. On the assumption that the algorithm for computing $f(A)$ requires $O(n^3)$ operations, the computation of $f(X_k)$ requires $O(8^k n^3)$ operations since the dimension of X_k is $2^k n \times 2^k n$ [18]. However, the computation of $f(X_{k-1})$ with $X_0 = A + \mathrm{i}hE_k$ requires $O(4 \cdot 8^{k-1} n^3)$, bearing in mind that the computational cost of complex arithmetic is about four times the computational cost of real arithmetic. Thus, the cost of $f(X_{k-1})$ with complex arguments is about half the cost of $f(X_k)$ for real arguments. Our numerical experiments below support this analysis.

In fact, we can evaluate the kth Fréchet derivative using the block matrix X_{k-2} instead of X_k. The idea is to use a mixed derivative scheme as shown below.

Lemma 3.1 *Suppose f, A, and E_j, $j = 1\colon k$, satisfy the assumptions of Theorem 2.1. Let $B = A + \mathrm{i}h_1 E_{k-1}$ then*

$$L_f^{(k)}(A, \mathcal{E}_k) = \lim_{(h_1,h_2)\to(0,0)} \frac{\operatorname{Im}\big(L_f^{(k-2)}(B + h_2 E_k, \mathcal{E}_{k-2}) - L_f^{(k-2)}(B - h_2 E_k, \mathcal{E}_{k-2})\big)}{2h_1 h_2}.$$

Proof Using (3.3) and (3.6), we obtain the mixed derivative

$$\begin{aligned} L_f^{(k)}(A, \mathcal{E}_k) &= \frac{L_f^{(k-1)}(A + h_2 E_k, \mathcal{E}_{k-1}) - L_f^{(k-1)}(A - h_2 E_k, \mathcal{E}_{k-1})}{2h_2} + O(h_2^2) \\ &= \left(\frac{\operatorname{Im}\big(L_f^{(k-2)}(B + h_2 E_k, \mathcal{E}_{k-2})\big)}{2h_1 h_2} - \frac{\operatorname{Im}\big(L_f^{(k-2)}(B - h_2 E_k, \mathcal{E}_{k-2})\big)}{2h_1 h_2} + O(h_1^2) \right) + O(h_2^2) \\ &= \lim_{(h_1,h_2)\to(0,0)} \frac{\operatorname{Im}\big(L_f^{(k-2)}(B + h_2 E_k, \mathcal{E}_{k-2}) - L_f^{(k-2)}(B - h_2 E_k, \mathcal{E}_{k-2})\big)}{2h_1 h_2}. \end{aligned}$$

Thus

$$L_f^{(k)}(A, \mathcal{E}_k) \approx \frac{\operatorname{Im}\big(L_f^{(k-2)}(A + \mathrm{i}h_1 E_{k-1} + h_2 E_k, \mathcal{E}_{k-2}) - L_f^{(k-2)}(A + \mathrm{i}h_1 E_{k-1} - h_2 E_k, \mathcal{E}_{k-2})\big)}{2h_1 h_2} \tag{3.10}$$

for sufficiently small real scalars h_1 and h_2. □

We can take advantage of the scheme (3.10) to reduce the computational cost for computing $L_f^{(k)}(A, \mathcal{E}_k)$, where $k \geq 3$. Consider the recurrence (2.3) for $X_0 = A + \mathrm{i}h_1 E_{k-1} + h_2 E_k$ and evaluate X_{k-2}, then set $X_0 = A + \mathrm{i}h_1 E_{k-1} - h_2 E_k$ and denote the value of X_{k-2} by Y_{k-2}. Observe that the top right $n \times n$ blocks of $f(X_{k-2})$ and $f(Y_{k-2})$ are $L_f^{(k-2)}(A + \mathrm{i}h_1 E_{k-1} + h_2 E_k, \mathcal{E}_{k-2})$ and $L_f^{(k-2)}(A + \mathrm{i}h_1 E_{k-1} - h_2 E_k, \mathcal{E}_{k-2})$, respectively. Therefore, we have

$$L_f^{(k)}(A, \mathcal{E}_k) \approx \operatorname{Im} \frac{[f(X_{k-2})]_{1n} - [f(Y_{k-2})]_{1n}}{2h_1 h_2}. \tag{3.11}$$

We will see in our numerical experiments that the parameter h_1 can be chosen as small as desired. However, the parameter h_2 is a finite difference step and it has to be chosen carefully. However, this approximation reduced the computational cost significantly. It requires two matrix function evaluations at complex matrices of size $2^{k-2}n \times 2^{k-2}n$, so the cost is $O(8 \cdot 8^{k-2}n^3)$, which is about *one-eighth* the cost of $f(X_k)$. Our numerical experiment below reveals this factor.

3.3 Exploiting action of matrix functions

Some matrix functions have existing algorithms to compute their actions on thin matrices without explicitly forming $f(X)$, but rather computing $f(X)B$ using the matrix–matrix product of X and B. As examples, the matrix exponential has the algorithm [7, Alg. 3.2] by Al-Mohy and Higham to compute $e^X B$. Recently, Al-Mohy [3] and Higham and Kandolf [16] developed algorithms to compute the actions of trigonometric and hyperbolic matrix functions. These Algorithms use truncated Taylor series, so $f(X)B$ is recovered via updating the matrix B of the product XB. Krylov subspace methods can also be used to evaluate matrix function times vector [15, Ch. 13].

As mentioned above, the kth Fréchet derivative is obtained by reading off the upper right $n \times n$ block of the matrix $f(X_k)$, where $X_0 = A$. This is equivalent to reading off the upper $n \times n$ block of the thin and tall matrix $f(X_k)B$, where

$$B = [0_n, 0_n, \ldots, 0_n, I_n]^T \in \mathbb{R}^{2^k n \times n}. \tag{3.12}$$

The entries 0_n and I_n are the zero and the identity matrices of size $n \times n$, respectively. The action of $f(X_k)$ on B extracts the last block column of $f(X_k)$. Thus,

$$L_f^{(k)}(A, \mathcal{E}_k) = [f(X_k)B]_{11}. \tag{3.13}$$

The advantage of this approach is that we only compute the rightmost n columns of the matrix $f(X_k)$ instead of computing the whole matrix as the algorithm of Higham and Relton does [18, Algorithm 3.6]. In addition, the matrix X_k has special structure, so the best algorithm is the one that exploits the structure of the input matrix in an optimal way. Such an algorithm is unavailable yet as per our knowledge. However, evaluating $f(X_k)B$ using the matrix multiplication $X_k B$ takes advantage of the sparsity of X_k. From (2.3) we count the nonzero elements of X_k, $\texttt{nnz}(X_k)$, in terms of $\texttt{nnz}(A)$ and $\texttt{nnz}(E_k)$. Thus, we have

$$\texttt{nnz}(X_k) = 2^k \texttt{nnz}(A) + 2^{k-1} \sum_{i=1}^{k} \texttt{nnz}(E_i). \tag{3.14}$$

Since $\texttt{nnz}(A)$ and $\texttt{nnz}(E_k)$ never exceed n^2, the nonzero elements of X_k is bounded above by $2^{k-1}(k+2)n^2$. To illustrate, for $k = 6$, $\texttt{nnz}(X_k)$ represents about 6.25% of the elements of X_k and this percentage drops down rapidly as k increases.

In the numerical experiment below, the methods that compute the kth Fréchet derivative s via the action of matrix functions are significantly faster (as k increases) than the methods that explicitly form the matrix $f(X_k)$ and then extract the top right $n \times n$ block.

Using the complex step approximation, we can reduce the size of the acting matrix as shown in Theorem 3.1. Thus for $X_0 = A + ihE_k$ and B is being reduced to size $2^{k-1}n \times n$ by deleting the first block, we have

$$L_f^{(k)}(A, \mathcal{E}_k) = \lim_{h \to 0} \frac{\operatorname{Im}[f(X_{k-1})B]_{11}}{h}. \tag{3.15}$$

We can go further and compute the action of the kth Fréchet derivative on a vector or more generally the action on a thin matrix b of size $n \times n_0$, where $n_0 \ll n$. Observe that we can compute $L_f^{(k)}(A, \mathcal{E}_k)b$ via (3.13) by multiplying its sides from the right by b. However, this would not be an efficient approach. The most efficient approach in this setting is to reconstruct the matrix B in (3.13) to be

$[0_{n\times n_0}, 0_{n\times n_0}, \ldots, 0_{n\times n_0}, b]^T =: B_b$, which is of size $2^k n \times n_0$. Thus,

$$L_f^{(k)}(A, \mathcal{E}_k)b = [f(X_k)B_b]_{11}. \tag{3.16}$$

By using the complex step for $X_0 = A + \mathrm{i}hE_k$ and adjusting B_b to be of size $2^{k-1}n \times n_0$ by deleting the top block, we obtain

$$L_f^{(k)}(A, \mathcal{E}_k)b = \lim_{h\to 0} \frac{\mathrm{Im}[f(X_{k-1})B_b]_{11}}{h}. \tag{3.17}$$

Kandolf and Relton [20] propose an algorithm for computing the action of the first Fréchet derivative $L_f(A, E)b$. Their approach is a particular case of (3.16) for $k = 1$; that is,

$$X_1 = \begin{bmatrix} A & E \\ 0 & A \end{bmatrix}, \quad f(X_1)\begin{bmatrix} 0 \\ b \end{bmatrix} = \begin{bmatrix} L_f(A, E)b \\ f(A)b \end{bmatrix}.$$

In fact, the computation of $L_f(A, E)b$ could be made more efficient if we use the complex step approximation

$$L_f(A, E)b \approx \mathrm{Im} f(A + \mathrm{i}hE)b/h,$$

which requires the action of $n \times n$ matrices instead of the action of $2n \times 2n$ matrices. This approximation is a particular case of (3.17) with $k = 1$. In similar fashions, we can implement (3.11) to reduce the size of the acting matrix to $2^{k-2}n \times 2^{k-2}n$ for $k \geq 3$.

4 Numerical experiments

In this section, we give several numerical experiments to illustrate the advantage of the use of the complex step approximation and the actions of matrix functions to reduce the computational cost. We use MATLAB R2017a on a machine with Core i7. For experiments where CPU time is important, we limit MATLAB to run on a single processor.

Experiment 1 In this experiment we measure the execution time for computing $L_f^{(k)}(A, \mathcal{E}_k)$, $k \geq 2$, by the methods prescribed above. We denote by $T_k^{(j)}$ the execution time for $L_f^{(k)}(A, \mathcal{E}_k)$ using the method j. We specify f to the matrix exponential function whose implementation in MATLAB is `expm`. This function is based on the algorithm of Al-Mohy and Higham [5, Algorithm 5.1]. We take `A = gallery('lesp',5)` and generate ten random matrices E_k, $k = 1\colon 10$, of size 5×5. We use the algorithm of Higham and Relton [18, Algorithm 3.6] that forms X_k in (2.3) with $X_0 = A$, computes $f(X_k)$, and extracts $L_f^{(k)}(A, \mathcal{E}_k)$ as $[f(X_k)]_{1n}$. The execution time for this method is denoted by $T_k^{(1)}$. Second, we run the algorithm of Higham and Relton using the initial matrix $X_0 = A + \mathrm{i}hE_1$ with $h = 10^{-8}$, and the direction matrices E_k for $k = 2\colon 10$. The algorithm computes $f(X_{k-1})$, of which $L_f^{(k)}(A, \mathcal{E}_k) \approx \mathrm{Im}[f(X_{k-1})]_{1n}/h$ as shown in (3.9). We denote the execution time for this implementation by $T_k^{(2)}$. Third, we use (3.11) and invoke the algorithm of

Higham and Relton for $f(X_{k-2})$ and $f(Y_{k-2})$ in the directions E_k, $k = 3\colon 10$, with initial matrices $X_0 = A + ih_1E_1 + h_2E_2$ and $Y_0 = A + ih_1E_1 - h_2E_2$, respectively. We take $h_2 = 10^{-6}$. We denote the execution time for this implementation by $T_k^{(3)}$. Fourth, we evaluate $L_f^{(k)}(A, \mathcal{E}_k)$ via (3.13) using the algorithm of Al-Mohy and Higham [7, Alg. 3.2], expmv, that computes the action $e^{X_k}B$, where X_k is given as in (2.3) with $X_0 = A$ and B is defined as in (3.12). The code is available at https://github.com/higham/expmv. We denote the running time for expmv by $T_k^{(4)}$. Fifth, we use expmv with complex step to approximate $L_f^{(k)}(A, \mathcal{E}_k)$ via (3.15). The running time for this implementation is denoted by $T_k^{(5)}$. Finally, we use (3.11) with expmv and denote its execution time by $T_k^{(6)}$.

Table 1 presents $T_k^{(j)}$ for $k \geq 2$ and $j = 1\colon 6$ and Fig. 1 plots the ratios $T_k^{(1)}/T_k^{(j)}$ and $j = 2\colon 6$. We fix $T_k^{(1)}$ as reference points to show how our approaches of using the complex step and matrix function actions improve on the approach of Higham and Relton. The results are as follows. We observe that the ratio $T_k^{(1)}/T_k^{(2)}$, $k \geq 2$ asymptotically approaches 2 as k increases. This supports our cost analysis given right after Theorem 3.2 that the use of the complex step approximation could save about 50% of the computational cost of the Algorithm of Higham and Relton. A superior computational saving is obtained when implementing the mixed derivatives. The ratio $T_k^{(1)}/T_k^{(3)}$, $k \geq 3$ approaches 8 as k increases, which also supports our cost analysis presented right after Lemma 3.1. The ratios $T_k^{(1)}/T_k^{(j)}$ for $j = 4, 5, 6$, where that actions of the matrix exponential are used, grow up linearly by a factor of 4. The implementation of expmv outperforms the direct use of expm for $k \geq 6$ when the dimensions of the input matrices start to grow up rapidly. expmv fully exploits the sparsity of the matrix X_k whereas the algorithm of expm involves explicit matrix powering and solves multiple right-hand sides linear systems, so the sparsity pattern deteriorates significantly. In addition, most of the components of expm(X_k) are not wanted. The three implementations of expmv behave in a similar manner with slight

Table 1 CPU time for computing $L_f^{(k)}(A, \mathcal{E}_k)$, $k = 2\colon 10$

k	$T_k^{(1)}$	$T_k^{(2)}$	$T_k^{(3)}$	$T_k^{(4)}$	$T_k^{(5)}$	$T_k^{(6)}$
2	8.79e−4	9.02e−4	—	8.77e−3	3.59e−3	—
3	1.44e−3	1.12e−3	1.37e−3	9.76e−3	9.32e−3	7.09e−3
4	3.32e−3	2.36e−3	2.19e−3	1.19e−2	1.07e−2	1.82e−2
5	1.26e−2	7.44e−3	4.53e−3	1.43e−2	1.35e−2	2.11e−2
6	8.20e−2	3.99e−2	1.46e−2	2.38e−2	1.94e−2	2.65e−2
7	6.15e−1	3.14e−1	8.63e−2	4.14e−2	3.67e−2	3.75e−2
8	4.69e0	2.40e0	6.27e−1	7.47e−2	6.67e−2	7.30e−2
9	3.68e1	1.85e1	4.81e0	1.42e−1	1.29e−1	1.34e−1
10	2.91e2	1.46e2	3.70e1	2.97e−1	2.61e−1	2.59e−1

For $j = 1\colon 3$, $T_k^{(j)}$ is the running time for direct computation of $f(X_{k-j+1})$ whereas $T_k^{(j+3)}$ is the running time for computing the action of $f(X_{k-j+1})$ on B

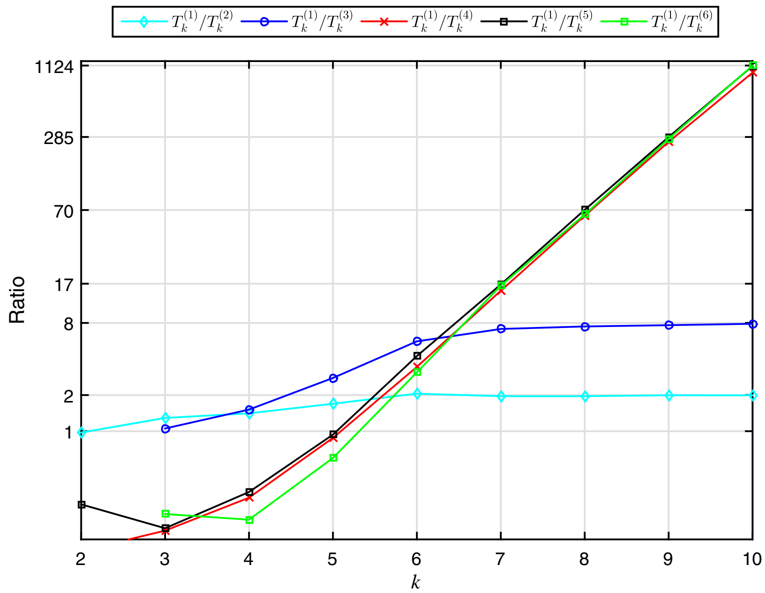

Fig. 1 The ratios $T_k^{(1)}/T_k^{(j)}$ for $k = 2\colon 10$ and $j = 2\colon 6$ of the data in Fig. 1

better performance of method 5 and method 6 that implement the complex step and the mixed derivative approximations, respectively.

We look at the accuracy of these methods. For each k we calculate 1-norm relative errors for $L_f^{(k)}(A, \mathcal{E}_k)$ where the "exact" reference computation is considered to be the output of the algorithm of Higham and Relton with `expm`. Figure 2 displays the results.

Note that the relative errors for $L_f^{(k)}(A, \mathcal{E}_k)$ computed using the complex step approximation (3.9) (err_1), the formula of matrix function action (3.13) (err_3), and

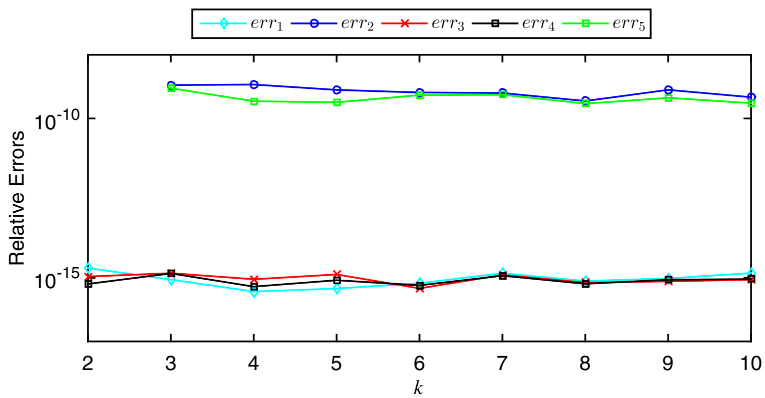

Fig. 2 Relative errors for the methods prescribed in Experiment 1

the complex step approximation with the matrix function action (3.15) (err_4) yield the highest accuracy whereas the use of the mixed derivative scheme (3.11) in both implementations produces less accurate computations (err_2 and err_5).

The accuracy of the mixed derivative scheme is quit sensitive to changes in the parameter h_2. The scheme is prone to subtractive cancellation in floating point arithmetic. Thus, h_2 has to be chosen to balance truncation errors with errors due to subtractive cancellation. Hence, the smallest relative error that could be obtained is of order $u^{1/2}$, where u is the unit roundoff [15, Section 3.4]. However, the complex step approximation does not involve subtraction. Thus, the parameter h_1 can be taken arbitrary small without effecting the obtained accuracy. The next experiment illustrates these points.

Experiment 2 The purpose of this experiment is to shed light on the robustness of the complex step approximation accuracy and some weakness on the mixed derivative approach. We take the matrix A and E_k from the above experiment and $f = \cos \circ \sqrt{\cdot}$. We use the algorithm of Al-Mohy [3, Algorithm 5.1] that computes the action of the matrix function $\cos(\sqrt{A})$ on a thin tall matrix B. The MATLAB code of this algorithm is denoted by `funmv` and is available at https://github.com/aalmohy/funmv. The algorithm is intended for half, single, and double precision arithmetics, but for this experiment we choose double precision. We compute $L := L_f^{(10)}(A, \mathcal{E}_{10})$ using (3.13) and consider it as a reference computation. Then, we compute L via complex step approximation with steps $h_1 = 10^{-r}$, $r = 1\colon 50$. The top part of Fig. 3 shows the 1-norm relative errors for each h_1. Note that the algorithm achieved the order of machine precision for all $r \geq 6$; this demonstrates the main advantage of complex

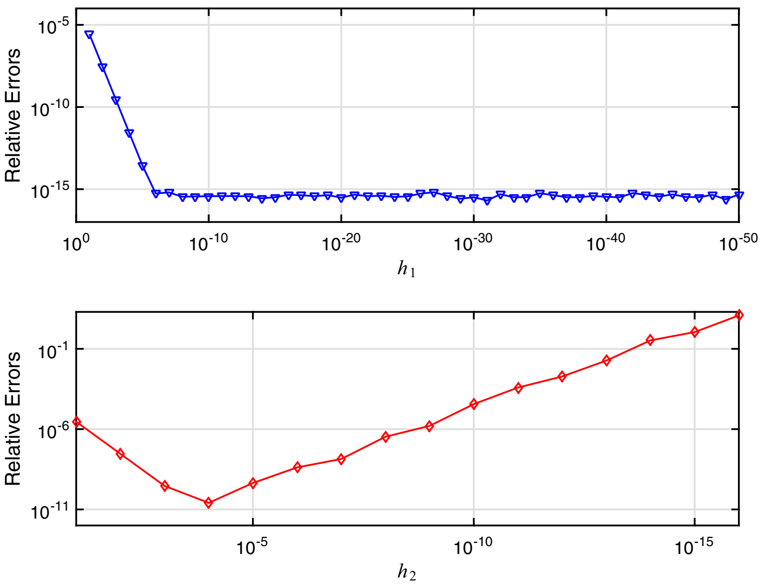

Fig. 3 The change of the relative error in (3.11) with the choice of h_1 and h_2

step approximations of derivatives. The bottom plot of Fig. 3 shows the sensitivity of the mixed derivative scheme (3.11) to the difference step h_2. We fixed $h_1 = 10^{-50}$ and take $h_2 = 10^{-r}$, $r = 1: 16$. Notice that the relative error decreases and attains its minimum at $r = 4$ then the accuracy deteriorates as r increases. The selection of an optimal r is a delicate matter; it depends on the inputs and the method by which f is computed.

5 Concluding remarks

The evaluation of a higher order Fréchet derivative of a matrix function is still a challenging problem if the algorithm that computes the matrix function f produces $f(X_k)$ as a full matrix. The dimension of the matrix X_k, $2^k n \times 2^k n$, grows up exponentially as k increases. The algorithm of Higham and Relton computes the kth Fréchet derivative for general complex functions and matrices. However, it does not exploit the special structure and sparsity pattern of the matrix X_k. We improve the efficiency of the computation of the kth Fréchet derivative in two ways. We use the complex step approximation to reduce the size of the problem and the action of matrix functions to exploit the structure of the problem. The application of the complex step approximation works for real-valued functions at real arguments. Our use of the complex step approximation reduces the dimension of the input matrix by half. The use of the mixed derivative scheme, however, reduces the dimension of the input matrix by three quarters. These implementations lead to significant computational savings compared to the direct use of X_k as proposed by Higham and Relton. Though the mixed derivative approach proves computational efficiency, it has to be used with caution since subtractive cancellation is likely to occur. The finite difference step has to be chosen to balance truncation errors with rounding errors. The complex step approximation is a reliable approach as the accuracy is retained whenever the complex step parameter becomes smaller.

Acknowledgments We thank the reviewers for their insightful comments and suggestions that helped to improve the presentation of this paper.

Funding information This work received funding from the Deanship of Scientific Research at King Khalid University through Research Groups Program under Grant No. R.G.P.1/113/40

References

1. Abreu, R., Stich, D., Morales, J.: On the generalization of the complex step method. J. Comput. Appl. Math. **241**, 84–102 (2013)
2. Ahipasaoglu, S.D., Li, X., Natarajan, K.: A convex optimization approach for computing correlated choice probabilities with many alternatives. IEEE Trans. Automat. Control **64**(1), 190–205 (2019)
3. Al-Mohy, A.H.: A truncated taylor series algorithm for computing the action of trigonometric and hyperbolic matrix functions. SIAM J. Sci. Comput. **40**(3), A1696–A1713 (2018)
4. Al-Mohy, A.H., Higham, N.J.: Computing the Fréchet, derivative of the matrix exponential, with an application to condition number estimation. SIAM J Matrix Anal. Appl. **30**(4), 1639–1657 (2009)
5. Al-Mohy, A.H., Higham, N.J.: A new scaling and squaring algorithm for the matrix exponential. SIAM J. Matrix Anal. Appl. **31**(3), 970–989 (2009)

6. Al-Mohy, A.H., Higham, N.J.: The complex step approximation to the Fréchet derivative of a matrix function. Numer. Algorithms **53**(1), 133–148 (2010)
7. Al-Mohy, A.H., Higham, N.J.: Computing the action of the matrix exponential, with an application to exponential integrators. SIAM J. Sci Comput. **33**(2), 488–511 (2011)
8. Al-Mohy, A.H., Higham, N.J., Relton, S.D.: Computing the Fréchet derivative of the matrix logarithm and estimating the condition number. SIAM J. Sci. Comput. **35**(4), C394–C410 (2013)
9. Amat, S., Busquier, S., Gutiérrez, J.M.: Geometric constructions of iterative functions to solve nonlinear equations. J. Comput. Appl. Math. **157**(1), 197–205 (2003)
10. Arioli, M., Benzi, M.: A finite element method for quantum graphs. IMA J. Numer. Anal. **38**(3), 1119–1163 (2018)
11. Benzi, M., Estrada, E., Klymko, C.: Ranking hubs and authorities using matrix functions. Linear Algebra Appl. **438**(5), 2447–2474 (2013)
12. Cardoso, J.A.R.: Computation of the matrix pth root and its Fréchet derivative by integrals. Electron. Trans. Numer. Anal. **39**, 414–436 (2012)
13. Cox, M.G., Harris, P.M.: Numerical analysis for algorithm design in metrology technical report 11, software support for metrology best practice guide, national physical laboratory, Teddington, UK (2004)
14. García-Mora, B., Santamaría, C., Rubio, G., Pontones, J.L.: Computing survival functions of the sum of two independent Markov processes: an application to bladder carcinoma treatment. Internat. J. Comput. Math. **91**(2), 209–220 (2014)
15. Higham, N.J.: Functions of matrices: theory and computation. society for industrial and applied mathematics, Philadelphia, PA USA (2008)
16. Higham, N.J., Kandolf, P.: Computing the action of trigonometric and hyperbolic matrix functions. SIAM J. Sci. Comput. **39**(2), A613–A627 (2017)
17. Higham, N.J., Lin, L.: An improved Schur–Padé algorithm for fractional powers of a matrix and their fréchet derivatives. SIAM J. Matrix Anal. Appl. **34**(3), 1341–1360 (2013)
18. Higham, N.J., Relton, S.D.: Higher order Fréchet derivatives of matrix functions and the level-2 condition number. SIAM J. Matrix Anal. Appl. **35**(3), 1019–1037 (2014)
19. Jeuris, B., Vandebril, R., Vandereycken, B.: A survey and comparison of contemporary algorithms for computing the matrix geometric mean. Electron. Trans. Numer. Anal. **39**, 379–402 (2012)
20. Kandolf, P., Relton, S.D.: A block Krylov method to compute the action of the Fréchet derivative of a matrix function on a vector with applications to condition number estimation. SIAM J. Sci. Comput. **39**(4), A1416–A1434 (2017)
21. Kenney, C.S., Laub, A.J.: A Schur–Fréchet algorithm for computing the logarithm and exponential of a matrix. SIAM J. Matrix Anal. Appl. **19**(3), 640–663 (1998)
22. Lai, K.-L., Crassidis, J.: Extensions of the first and second complex-step derivative approximations. J. Comput. Appl. Math. **219**(1), 276–293 (2008)
23. Lai, K.-L., Crassidis, J., Cheng, Y., Kim, J.: New complex-step derivative approximations with application to second-order Kalman filtering. AIAA Guidance, Navigation, and Control Conference and Exhibit, 5944 (2005)
24. Lantoine, G., Russell, R.P., Dargent, T.: Using multicomplex variables for automatic computation of high-order derivatives. ACM Trans. Math Softw. **38**, 16:1–16:21 (2012)
25. Lyness, J.N.: Numerical algorithms based on the theory of complex variable. In: Proceedings of the 1967 22-nd National Conference, Washington, D.C. USA, pp. 125–133 (1967)
26. Lyness, J.N., Moler, C.B.: Numerical differentiation of analytic functions. SIAM J. Numer. Anal. **4**(2), 202–210 (1967)
27. Noferini, V.: A formula for the Fréchet derivative of a generalized matrix function. SIAM J. Matrix Anal. Appl. **38**(2), 434–457 (2017)
28. Powell, S., Arridge, S.R., Leung, T.: Gradient-based quantitative image reconstruction in ultrasound-modulated optical tomography: first harmonic measurement type in a linearised diffusion formulation. IEEE Trans. Med. Imag. 35 (2015)
29. Rossignac, J., Vinacua, A.: Steady affine motions and morphs. ACM T. Graphic **30**, 116:1–116:16 (2011)
30. Squire, W., Trapp, G.: Using complex variables to estimate derivatives of real functions. SIAM Rev. **40**(1), 110–112 (1998)